# Automated Retrieval of Semantic Web Services: A Matching Based on Conceptual Indexation

Hadjila Fethallah and Chikh Mohammed Amine
Computer Science Department, UABT University Algeria

**Abstract:** *Web services are taking an important place in the distributed computing field, as well as in the electronic business. In this paper we present an initial research which deals with the issue of automated service retrieval. For that, we propose an approach that exploits the service interface (inputs/outputs) and the domain ontology, in order to conceptually index the web services. After that we compute a similarity score between the request and the indexed web services through the cosine measure. An experimentation based on the OWLTC test collection is described to evaluate the system. The obtained results are very encouraging and confirm the suitability of the solution.*

**Keywords:** *Web service research, similarity measure, ontologies, OWLS, information retrieval.*

*Received October 17, 2010; accepted January 3, 2011*

## 1. Introduction

The web service technology is a concrete instance of the SOA paradigm. It is important to mention that the majority of the existing web applications is planned to undergo a reengineering process [8], in order to take advantages of the SOA, which include reusability, interoperability, integration etc. The web services are based on a set of recommendations that include Universal Discovery Description Integration (UDDI) [27], Web Services Description Language (WSDL) [10] and Simple Object Access Protocol (SOAP) [28].

SOAP is designed to provide a transport mechanism for the XML based message. WSDL model is used to describe the service interface. However, neither SOAP nor WSDL allow automated location of web services on the basis of their capabilities. UDDI is a registry that describes businesses by their characteristics such as names and other public information. The registry also describes the services through the industrial categories. In addition, UDDI descriptions are augmented by a set of attributes called Tmodels, which describe additional features such as classification of services within taxonomies and abstract specifications, etc. Since UDDI does not represent service capabilities, it cannot help the functional search for services. In addition to the limitations mentioned above, XML based standards lack the explicit semantics, that is two identical XML descriptions could mean totally different things, depending on the context of their use. The opposite case is also possible (i.e, the presence of two XML documents that have same meaning and different structures). These shortcomings limit the capability of matching web services. To resolve this problem, we must add semantic knowledge to support the identification of the most suitable service for a particular task. The integration effort of semantics into web services started with the RDF language [20] and evolved with the creation of RDFS [9], OWL [14] and OWLS [15]. Different approaches were developed to resolve the web service searching problem [5, 16, 19, 25]. In this paper we propose, an approach based on the OWLS content we exploit, more precisely the “profile: hasInput” and the “profile: hasOutput” elements to index the services and the requests. Furthermore we employ a set of domain ontologies to enhance the semantic of these elements. This process is called “conceptual indexation”, it is described in the fourth section. The searching process of a web service can be summarized as follow:

- For each web service (S) of the base, we build 02 concepts vectors (Si) and (So) which model (S) in the matching step.
- Si contains the service inputs and their subsumers, which are extracted from the domain ontology. It was noticed that each subsumer is associated with its frequency. For that we use the domain ontology associated with S.
- So is the same as S, except that it works on the outputs, rather than the inputs.
- The same thing is applied to the requests to generate the two resultant vectors Ri and Ro (a request is modeled as an OWLS document).
- The matching process is implemented by the cosine measure which computes a similarity score between the service vectors and the request vectors.
- A user-fixed threshold θ is used to filter the results, according to their closeness degree.
- Finally we evaluate the system quality through the recall and the precision metrics. We have chosen the measure cosine (which is a space vector similarity measure), because it is one of the most prominent

measures of the information retrieval domain [11].

The rest of the paper is structured as follows: Section 2 presents a background on service retrieval approaches. Section 3 presents the OWLS specification. In section 4 we present our proposal for the web service retrieval problem. Section 5 shows the experimental results; we discuss the obtained results in section 6. Finally, in section 7 we expose some open issues and conclude this paper.

## 2. State of The Art

Different web service matchmakers have been developed in the literature, such as the MAMA [12], HotBlu [13, 18], OWLS-MX [19], RACER [21], SDS [22], and OWLSUDDI matchmaker [25]. Many of them use the subsumption test and the description logics to compare the (I/O) of the profile with the user's request. We mention also another set of approaches [1, 7] which use only the WSDL interface as an entry (without semantic interfaces likes OWLS). The SecDisc approach [1] utilizes two algorithms: the level-based matching is used to compare syntactically the WSDL concrete parts, and the sequence-based schema matching approach [2] is used to compare the WSDL abstract parts, in the same optic, the approach quoted in [23] uses a linear discrimination function associated with the wordnet thesaurus [29], to compare a set of WSDL files.

The system proposed in [6] adopts a service process-model matching. To reinforce the retrieval scalability, the approach developed in [3] makes a peer-to-peer discovery of web services. Bansal and Vidal [4] apply a recursive tree matching to achieve the retrieval problem. The work in [17] proposes a set of mediators to make a semantic discovery of WSMO based web services, the matching is done through the Web Service Modeling Language (WSML). The METEOR-S approach [26] enhances the WSDL standard, with semantic descriptors in order to make an efficient search. LARKS [24] and OWLSMX [19] provide a hybrid search, in the sense that they exploit both explicit (formal) and implicit (informal) semantics by complementary means of logic based and approximate matching.

The OWLS-MX [19] matchmaker proposes four variants of the matching algorithm, the first is purely logic and the others are hybrids (i.e, they combine the content based metrics used in the information retrieval in addition to the logic subsumption). Each variant uses a set of filters and produces seven scores of matching.

The purely logic-based variant OWLS-M0 (the first algorithm) is similar to the OWLS-UDDI matchmaker [25] but with the following differences: Firstly, the latter adopts a different notion of plug-in matching and does not provide additional subsumed-by matching.

## 3. Ontology Web Language Services (OWLS)

OWLS [15] is a high level ontology, which allows an abstract description of a service. It is composed of a root class named "Service" and it directly corresponds to the actual service which is described semantically (every service that is described, corresponds to an instance of this concept). The "Service" class is linked with three other classes; the first is "ServiceProfile", it specifies the functional properties as well as the QOS based attributes of a service, it also gives an informal description about the exposed capabilities. The second is "ServiceModel", it is an orchestration part that specifies the data flow and the control flow of the service.

The third class is named "ServiceGrounding", its role is to define the manner to access a service. It also shows the equivalent element in the WSDL model, for each atomic process. Figure 1 Shows the upper ontology OWLS.

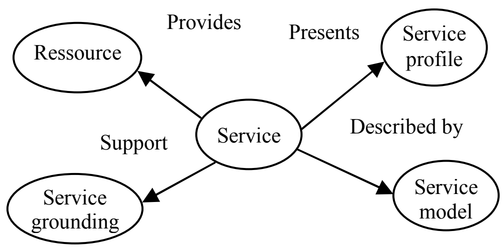


Figure 1. The upper ontology OWLS.

## 4. Contribution

The data used in our experiment is sampled from the OWLS-TC[1] corpus version 2.2.1, this base is developed by the German research center for artificial intelligence (http://www.dfki.de/scallops). The base uses the OWLS documents to describe a set of web services. These documents involve in their profile part, a set of elements named «profile: hasinput» and «profile: hasoutput». These elements are used during the conceptual indexation of the web services. The OWLS documents are divided into seven classes: economy, communication, education, food, weapon, medical care and travel. In this paper we have used only two requests (these requests, correspond to the classes that have the greatest number of documents).

We noticed that the request is modeled as a web service, i.e., it owns an OWLS document with a «profile: hasinput» and a «profile: hasoutput» elements.

The searching process follows these consecutive stages:

- *The Indexation Stage:*
  1. For each service, we extract its inputs and outputs.

[1] http://www.dfki.de/scallops

2. We build a vector Vi which contains the inputs and their subsuming concepts in the ontology (each concept will have a number which represents its frequency in the vector). For example if the inputs are {c1, c2}, where c1 and c2 contain the subsumers {c3, c4} and {c5, c4} respectively, then vi={c1/1, c2/1, c3/1, c4/2, c5/1} (because c4 has two occurrences).
3. The output (Vo) is computed in the same manner.
4. The same treatment of indexation (step 2 and 3) is applied to the request (we compute Ri and Ro).

- *The Matching Stage:*

1. We compare the indexed request with the indexed services by using the cosine similarity measure as given by the following formula: cos (A, B)=<A, B>/(||A||.||B||) where <A, B> denotes the scalar product of A and B. A and B are two conceptual vectors. This step is analyzed as follows:

   a. Score 1=cos (Ri, Si). Score1 stands for the similarity between Ri and Si.
   b. Score 2=cos (Ro, So). Score2 stands for the similarity between Ro and So.
   c. Then, Score=(Score 1+Score 2)/2.

2. We then sort the results from the greatest score to the weakest score.
3. We retain only the services whose score exceeds a certain threshold θ (the threshold θ is a numerical value chosen by the user, θ∈ [0, 1]).

Finally, we evaluate the results through the criteria of recall and precision.

## 5. Experimental Procedure

The following paragraph defines the criteria of recall and precision, which are used in the results evaluation. The precision P can be thought of as the 'signal to noise' ratio, it is the fraction between the pertinent results retrieved by the system and the total results. In other words $P=Tp/(Tp+Fp)$ where:
$Tp$ is the number of pertinent results which are positively classified
$Fp$ is the number of impertinent results which are positively classified.
The recalled $R$, can be thought of as the 'hit ratio'; it is the fraction between the pertinent results retrieved by the system and the total pertinent results, $R=Tp/(Tp+Fn)$ where:
$Fn$ is the number of pertinent results which are negatively classified. In this experiment, we consider two requests $R1$ and $R2$ shown in Table 1:

Table1. The experiment requests.

| Name | Inputs | Outputs |
|---|---|---|
| Car_price_service | Car | Price |
| Grocerysore_food_serveice | Grocerystore | Food |

Table 2 shows a part of the results associated with the first request $R1$, where θ is set to 0.8. The 7th column represents the decision taken by our system, while the 8th column represents the decision taken by the human expert. The human decision is given in the OWLTC collection, for each pair: request-web service.

Table 3 shows a part of the results associated with the second request R2, in this example θ is set to 0.8. The 7th column represents the decision taken by our system, while the 8th column represents the decision taken by the human experts.

Table 2. The results of R1.

| The Service Name | The Service Inputs | Score 1 | The Service Outputs | Score 2 | Score | The System Decision | The User's Decision |
|---|---|---|---|---|---|---|---|
| 3wheeledcar_price | 3wheeledcar | 0.953 | Price | 1.0 | 0.976 | Accept | Accept |
| 1personbicyclecar_price | _4WheeledCar, _1personbicycle | 0.809 | Price | 1.0 | 0.904 | Accept | Reject |
| car_price | Car | 1.0 | Price | 1.0 | 1.0 | Accept | Accept |
| 3wheeledcaryear_Recommendedprice | 3WheeledCar, year | 0.725 | Recommended Price | 0.5 | 0.612 | Reject | Accept |
| _3WheeledAudiCarprice_ | - | 0.0 | Price | 1.0 | 0.5 | Reject | Accept |
| _food_Exportservice | - | 0.0 | Food | 0.0 | 0.0 | Reject | Reject |
| drugstore tea | Drugstore | 0.463 | Tea | 0.0 | 0.231 | Reject | Reject |
| grocerystore_butterquantity | Grocery store | 0.463 | Butter, quantity | 0.0 | 0.231 | Reject | Reject |
| grocerystore_food_service | Grocery store | 0.463 | Food | 0.0 | 0.231 | Reject | Reject |
| retailstore_foodquality_service | Retail store | 0.467 | Food, quality | 0.0 | 0.233 | Reject | Reject |

Table 3. The results of R2.

| The Service Name | The Service Inputs | Score 1 | The Services Outputs | Score 2 | Score | The System Decision | The User's Decision |
|---|---|---|---|---|---|---|---|
| 3wheeledcar_price | 3wheeledcar | 0.442 | Price | 0.0 | 0.221 | Reject | Reject |
| 1personbicyclecar_price | _4WheeledCar, _1personbicycle | 0.625 | Price | 0.0 | 0.312 | Reject | Reject |
| car_price | Car | 0.463 | Price | 0.0 | 0.231 | Reject | Reject |
| 3wheeledcaryear_Recommendedprice | 3WheeledCar, year | 0.336 | Recommended price | 0.0 | 0.168 | Reject | Reject |
| 3WheeledAudiCarprice_ | - | 0.0 | Price | 0.0 | 0.0 | Reject | Reject |
| _food_Exportservice | - | 0.0 | Food | 1.0 | 0.5 | Reject | Reject |
| drugstore tea | Drugstore | 0.985 | Tea | 0.932 | 0.958 | Accept | Accept |
| grocerystore_butterquantity | Grocery store | 1.0 | Butter, quantity | 0.745 | 0.872 | Accept | Accept |
| grocerystore_food_service | Grocery store | 1.0 | Food | 1.0 | 1.0 | Accept | Accept |
| retailstore_foodquality_service | Retail store | 0.992 | Food, quality | 0.912 | 0.952 | Accept | Accept |

## 6. Discussion

Figures 2 and 3 show the relation between the performance (recall/precision) and the threshold θ.

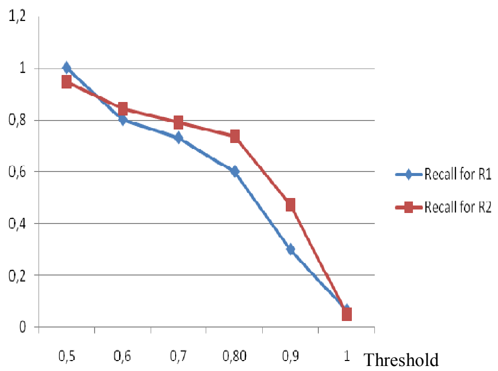


Figure 2. The system recall.

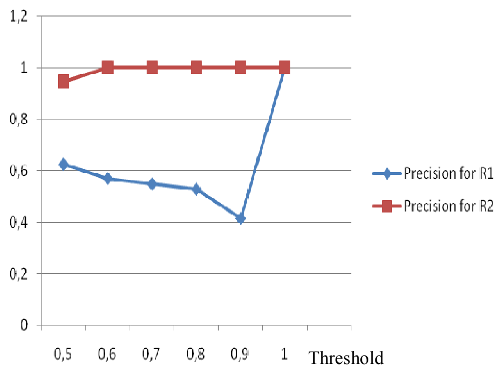


Figure 3. The system precision.

The bigger the threshold θ, the better the precision and the more mediocre the recall becomes. For R1 we notice that a threshold comprised between 0.5 and 0.9, induces a reduction in the precision and the recall. This situation is caused by the degradation of Tp and the increasing of fn. We also noticed that if θ=1, then the precision=1 and the recall is minimized.

For R2, we noticed an increasing in the precision, which will be equal to 1. This situation is caused by the elimination of Fp. Generally, it is noticed that the range [0.7, 0.8] is a good compromise (for the precision and the recall). The main difficulty of the approach is to set the threshold θ, this problem can be resolved by setting θ heuristically, or by learning the parameter from the precedent experiences (or the user's feedback).

Another drawback is the time needed to build the model, associated to the services/or the requests, this drawback has a strong link with the selected similarity measure. It is observed that the main advantages of the system, in comparison with the other approaches are:

1. The adaptation of the best space vector similarity measure to the semantic context.
2. The minimization of Fp and Fn: the adopted process of indexation eliminates the ambiguity in the computation of the similarity score, however the bipartite graph matching approaches such as [24], cannot easily bypass this problem.

## 7. Conclusions

In this paper, we have presented a method that exploits OWL-S to build a web service searching system. This method uses the service inputs/outputs and the ontologies to semantically index, the web services, then it exploits the cosine similarity measure to select the relevant solutions.

Future work will mainly consist of augmenting the requests number that have to be tested. Furthermore, an interesting aspect to deal with, is the study of the similarity measures performance, which will include the precision, the recall and the time.

## References

[1] Algergawy A., Schallehn E., and Saake G., "Combining Schema and Level-Based Matching for Web Service Discovery," *in Proceedings of the 10th International Conference of Web Engineering*, Heidelberg, pp. 114-128, 2010.

[2] Algergawy A., Schallehn E., and Saake G., "Improving XML Schema Matching Performance Using Prufer Sequences," *Data and Knowledge Engineering*, vol. 68, no. 8, pp. 728-747, 2009.

[3] Banaei-kashani F., Chen C., and Shahabi C., "WSPDS: Web Services Peer-to-Peer Discovery Service," *in Proceedings of International*

*Symposium on Web Services and Applications*, USA, pp. 733-743, 2004.

[4] Bansal S. and Vidal J., "Matchmaking of Web Services Based on the DAML-S Service Model," *in Proceedings of Second International Conference on Autonomous Agents and Multi-Agent Systems*, Australia, pp. 926-927, 2003.

[5] Benatallah B., Hacid M., Rey C., and Toumani F., "Semantic Reasoning for Web Services Discovery," *in Proceedings of 2nd International Semantic Web Conference*, USA, pp. 242-257, 2003.

[6] Bernstein A. and Klein M., "Towards High-Precision Service Retrieval," *IEEE Internet Computing*, vol. 8, no. 1, pp. 30-36, 2004.

[7] Bose A., Nayak R., and Bruza P., "Improving Web Service Discovery by Using Semantic Models," *in Proceedings of the 9th International Conference on Web Information Systems Engineering WISE*, in: Bailey J., Maier D., Schewe K.-D., Thalheim B., Wang X.S. (Eds.), Heidelberg, pp. 366-380, 2008.

[8] Bouchiha D., Malki M., and Benslimane S., "Towards Reengineering Web Applications to Web Services," *The International Arab Journal of Information Technology*, vol. 6, no. 4, pp. 359-364, 2009.

[9] Brickley D. and Guha R., (Eds.), "RDF Vocabulary Description Language 1.0: RDF Schema," *Working Draft, W3C*, available at: http://www.w3.org/rdf-schema, last visited 2003.

[10] Chinnici R., Gudgin M., Moreau J., and Weerawarana S., "Web Services Description Language (WSDL) Version 1.2," *W3C Working Draft*, available at: http: //www.w3.org/TR/ 2003/WD-wsdl12-20030124/, last visited 2003.

[11] Cohen W., Ravikumar, P., and Fienberg S., "A Comparison of String Distance Metrics for Name-Matching Tasks," *in Proceedings of Information Integration on the Web*, USA, pp. 73-78, 2003.

[12] Colucci S., Noia T., Sciascio E., Donini F., and Mongiello M., "Concept Abduction and Contraction for Semantic-Based Discovery of Matches and Negotiation Spaces in an E-Marketplace," *in Proceedings of the 6th International Conference on Electronic Commerce*, USA, pp. 41-50, 2004.

[13] Constantinescu I. and Faltings B., "Efficient Matchmaking and Directory Services," *in Proceedings of IEEE/WIC International Conference on Web Intelligence*, Canada, pp. 75-81, 2003.

[14] Dean M. and Schreiber G., (Eds.), "OWL Web Ontology Language: Reference," *W3C Working Draft*, available at: http://www.w3.org/TR/owl-ref/, last visited 2003.

[15] Dean M. (Eds.), "OWL-S: Semantic Markup for Web Services Version 1.0," *W3C Working Draft*, available at: http://www.daml.org/services/, last visited 2004.

[16] Hadjila F. and Chikh A., "Recherche Des Services Web en Utilisant le Contenu d'OWLS," *Revue Rist*, vol .18, no. 1, pp. 144- 156, 2010.

[17] Keller U., Lara R., Polleres A., and Fensel D., "Automatic Location of Services," *in Proceedings of European Semantic Web Conference*, Crete, pp. 1-16, 2005.

[18] Klein M. and Koenig-Ries B., "Coupled Signature and Specification Matching for Automatic Service Binding," *in Proceedings of European Conference on Web Services*, Germany, pp. 183-197, 2004.

[19] Klusch M., Fries B., Mahboob K., and Sycara K., "OWLS-MX: Hybrid OWL-S Service Matchmaking," *in Proceedings of 1st International AAAI Fall Symposium on Agents and the Semantic Web*, 2005.

[20] Klyne G. and Carroll J., (Eds.), "Resource Description Framework (RDF): Concepts and Abstract Syntax," *W3C Working Draft*, available at: http://www.w3.org/TR/rdf-concepts, last visited 2003.

[21] Li L. and Horrock I., "A Software Framework for Matchmaking Based on Semantic Web Technology," *in Proceedings of the 12th International World Wide Web Conference Workshop on E-Services and the Semantic Web*, USA, pp. 331-339, 2003.

[22] Mandell D. and McIllraith S., "A Bottom-up Approach to Automating Web Service Discovery, Customization, and Semantic Translation," *in Proceedings of the 12th International Conference on the World Wide Web*, Budapest, 2003.

[23] Sotolongo R., Kobashikawa C., Dong F., and Hirota K., "Algorithm for Web Service Discovery based on Information Retrieval using WordNet and Linear Discriminant Functions," *Journal of Advanced Computational Intelligence and Intelligent Informatics*, vol. 12, no. 2, pp. 182-189, 2008.

[24] Sycara K., Klusch M., Widoff S., and Lu J., "Larks: Dynamic Matchmaking among Heterogeneous Software Agents in Cyberspace," *Autonomous Agents and Multi-Agent Systems*, vol. 5, no. 2, pp. 173-203, 2002.

[25] Sycara K., Paolucci M., Anolekar A., and Srinivasan N., "Automated Discovery, Interaction and Composition of Semantic Web Services," *Web Semantics Science Services and Agents on the World Wide Web*, vol. 1, no. 1, pp. 27-46, 2003.

[26] Verma K., Sivashanmugam K., Sheth A., Patil A., Oundhakar S., and Miller J., "METEOR-S WSDI: A Scalable Infrastructure of Registries for

Semantic Publication and Discovery of Web Services," *Journal of Information Technology and Management*, vol. 6, no. 1, pp. 17-39, 2004.
[27] W3C "UDDI Technical White Paper," *W3C* available at: http://www.uddi.org/pubs/Iru_UDDI_Technical_White_Paper.pdf, last visited 2000.
[28] W3C "Recommendation SOAP Version 1.2," Part 0: Primer, available at: http://www.w3.org/TR/ 2003/REC-soap12-part0-20030624/, last visited 2003.
[29] WordNet a Lexical Database for the English Language, available at: http://wordnet. princeton.edu/, last visited.

**Hadjila Fethallah** received his MS degree in computer science from the UABT University of Tlemcen in 2003. Currently, he is a researcher in the Department of Computer Science, UABT University Tlemcen, Algeria. His research interests include: web service discovery, service composition, ontology matching, and artificial intelligence.

**Chikh Mohammed Amine** received his PhD degree in 2005. He became an Assistant Professor in the Department of Computer Science, UABT University of Tlemcen since 2006, His research interests include: decision aiding, artificial intelligence, biomedical applications.